

## PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2003-168284

(43)Date of publication of application : 13.06.2003

(51)Int.Cl. G11B 27/034  
G11B 20/10  
G11B 20/12  
G11B 27/00  
H04N 5/91

(21)Application number : 2001-363589

(71)Applicant : SHARP CORP

(22)Date of filing : 29.11.2001

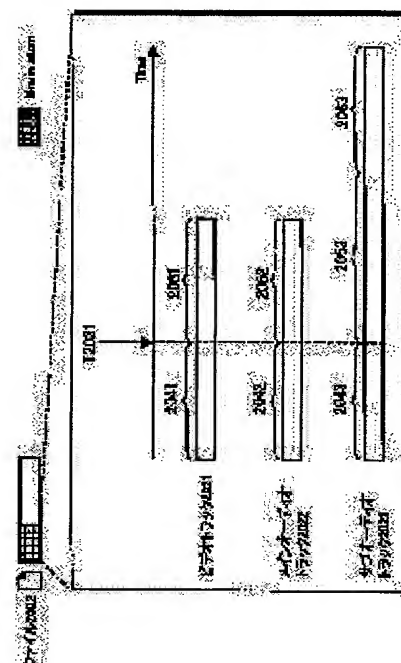
(72)Inventor : KIYAMA JIRO  
IWANO HIROTOSHI  
YAMAGUCHI TAKAYOSHI

## (54) DATA RECORDING METHOD AND DATA EDITING METHOD

## (57)Abstract:

**PROBLEM TO BE SOLVED:** To provide a method of saving user's trouble when editing a video to which a sub audio track is added.

**SOLUTION:** First data and second data are correlated and recorded. The second data are classified into a first class to be synchronized with a particular part of the first data and a second class not to be synchronized with a particular part of the first data, and the classified data are recorded. That is, information showing a (independent/dependent) relation of the sub audio track to a video track is prepared, and a treatment of the sub audio track in editing the video track is changed on the basis of the information.

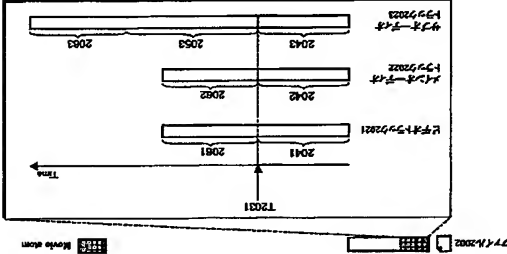


|                          |        |                             |       |          |       |   |       |  |  |
|--------------------------|--------|-----------------------------|-------|----------|-------|---|-------|--|--|
| (51)Int.Cl. <sup>7</sup> |        | 識別記号                        |       | FI       |       | チャート(参考)  |       |  |  |
| G11B                     | 27/034 | G11B                        | 20/10 | G        | 5C053 | 5D044   | 5D110 |  |  |
|                          | 20/10  |                             | 20/12 |          |       |   |       |  |  |
|                          | 20/12  |                             | 27/00 |          |       |   |       |  |  |
| H04N                     | 27/00  | H04N                        | 27/02 | K        | N     | N   | N     |  |  |
|                          | 5/91   |                             | 5/91  |          |       |   |       |  |  |
| 審査請求 未請求                 |        | 請求項の数 3                     |       | OL       |       | (全 22 頁)  |       |  |  |
| (21) 出願番号                |        | 特願2001-363568(P2001-363568) |       | (71) 出願人 |       | 000005049   |       |  |  |
| (22) 出願日                 |        | 平成13年11月29日(2001.11.29)     |       | (72) 発明者 |       | シャープ株式会社<br>大阪府大阪市阿倍野区長瀬町22番22号<br>木山 次郎<br>大阪府大阪市阿倍野区長瀬町22番22号 シ<br>シャープ株式会社内<br>岩野 裕利<br>大阪府大阪市阿倍野区長瀬町22番22号 シ<br>シャープ株式会社内<br>100102277<br>(74) 代理人 弁理士 佐々木 晴康 (外 2 名) |       |  |  |

(54) 【発明の名称】 データ記録方法およびビデオ編集方法

(57) 【要約】  
【課題】 サブオーディオの付加されたビデオを編集する際のユーザの手間を削減する方法を提供することを目的とする。

【解決手段】 第1のデータと第2のデータとを関係付けて記録する記録方法であって、前記第2のデータを、前記第1のデータの特定部分との同期が必要な第1の種別と、前記第1のデータの特定部分との同期が不要な第2の種別とに区別して記録する。すなわち、ビデオトラックに対するサブオーディオトラックの関係 (独立/依存) を示す情報を用意し、その情報に基づきビデオトラック編集時のサブオーディオトラックの扱いを変える。



【特許請求の範囲】

【請求項1】 第1のデータと第2のデータとを関係付けて記録する記録方法であって、

前記第2のデータを、前記第1のデータの特定部分との同期が必要な第1の種別と、前記第1のデータの特定部分との同期が不要な第2の種別とに区別して記録することを特徴とするデータ記録方法。

【請求項2】 前記請求項1に記載の記録方法に従って記録されたデータの編集方法であって、

前記第1のデータの編集に際して、前記第1の種別に区別された第2のデータは、関係する第1のデータと同時に編集することを特徴とするデータ編集方法。

【請求項3】 前記請求項1に記載の記録方法に従って記録されたデータの編集方法であって、

前記第1のデータの編集に際して、前記第1の種別に区別された第2のデータは、関係する第1のデータと同時に編集しないことを特徴とするデータ編集方法。

【発明の詳細な説明】  
【0001】

【発明の属する技術分野】 本発明は、ハードディスク、光ディスク等のランダムアクセス可能な記録媒体に対し、映像データ、音声データを記録・編集するデータ記録方法及びデータ編集方法に関するものである。

【0002】

【従来の技術】 ディスクメディアを用いたビデオのディジタル記録再生装置 (以下、ビデオディスクレコーダと呼ぶ) が普及しつつある。テープメディアにはないディジタルメディアにおける特徴機能として、非破壊編集機能あるいはノンリニア編集機能と呼ばれるものがある。この機能は、ディスク上に記録したAVストリームを移動あるいはコピーすることなく、AVストリームの任意の区間 (シーン) を任意の順番で再生できる、というもので、AVストリームのどこからどこまでという順番で再生するを示す情報 (再生管理情報) を作り、その情報に従って再生することで実現される。

【0003】 例えば、図30に示すように、あるコンテントがビデオデータと、前記ビデオデータと同時に収録し、前記ビデオデータと同期して再生されるべきオーディオデータと、前記ビデオデータおよびオーディオデータとは別に後から追加したオーディオデータとの3種類のデータで構成され、それぞれをトラックとして、Video track3000、Main audio track3010、Sub audio track3020の3トラックで管理していたとする。

【0004】 このとき、図中の区間T3051からT3052の区間を削除しようとした場合、ユーザの望むであろう結果として次の2種類が考えられる。第1の結果は、T3051からT3052の区間に含まれる3トラックの区間すべて (図中の区間3002、3012、3022) が再生対象からはずれ、図1のように、その前後の区間が連続的に再生されることである。

第2の結果は、再生対象からはずれた区間の前後の区間が連続的に再生されることである。

【0005】 このような結果は、再生対象からはずれない区間の関係が強い場合、例えばSub audio track3020の区間3022がVideo track3000の区間3002に対する英語での吹き替えであった場合に望まれると考えられる。なぜなら、Sub audio track3020の区間3022が再生対象から外されなかった場合、対応するビデオの無い英語の吹き替え音声のみが残るからである。

【0006】 一方、ユーザの望むであろう第2の結果は、T3051からT3052の区間に含まれるVideo track3000およびMain audio track3001の区間 (図中の区間3002、3012) のみ再生対象から外れ、図32のように、Video track3000およびMain audio track3010に関し、その前後の区間が連続的に再生され、Sub audio track3020に関しては削除前と変わらず再生されることである。

【0007】 このような結果は、Sub audio track3020のデータの連続性が高く、他のトラックとの関係が弱い場合、例えばSub audio track3020がVideo track3000に属する区間に含まれると望まれると考えられる。なぜなら、Sub audio track3020の区間3022が再生対象からはずれた場合、BGMが不連続になるためである。

【0008】 従来のテープメディアを用いた編集の場合、第1の結果を得るのには、コピーの際、T3051からT3052を除くだけでよいという比較的容易であるが、第2の結果を得ようとする、まず、T3051からT3052を除いてコピーし、再度Sub audio trackのデータを録音する必要がある。二度手間となる。それに対してディスクメディアでは、トラックの管理情報を書き換えるだけで第1、第2の結果を容易に得ることができる。

【0009】

【発明が解決しようとする課題】 しかしながら、従来の技術において、前記第1の結果を得ようとする、ユーザは図30に示すような画面から区間3002、3012、3022をそれぞれ選択し、削除コマンドを実行することになる。同様に、前記第2の結果を得ようとする、ユーザは図30に示すような画面から区間3002、3012を選択し、削除コマンドを実行することになる。すなわち、この削除コマンドを個別に明示的に指定する必要がある。

【0010】 本発明は、上記課題を鑑みてなされたものであり、サブオーディオの付加されたビデオを編集する際のユーザの手間を削減することが可能なデータ記録方法及びデータ編集方法を提案することを目的とする。

【0011】

【課題を解決するための手段】 本願の第1の発明は、第1のデータと第2のデータとを関係付けて記録する記録方法であって、前記第2のデータを、前記第1のデータの特定部分との同期が必要な第1の種別と、前記第1のデータの特定部分との同期が不要な第2の種別とに区別して記録することを特徴とする。

【0012】 本願の第2の発明は、前記第1のデータの編集に際して、前記第1の種別に区別された第2のデータ

が、関係する第1のデータと同時に編集されることを特徴とする。

【0013】本願の第3の発明は、前記第1のデータの編集に際して、前記第1の種類の類別に区別された第2のデータが、関係する第1のデータと同時に編集されないことを特徴とする。

【0014】  
【発明の実施の形態】以下、本発明の実施形態について、図面を参照しながら詳細に説明する。ここでの説明は、本発明において共通に用いる構成、個々の実施形態に固有の内容という順に行っていく。

【0015】<システム構成>図1は本発明において共通に用いる、アプリケーション可能なビデオディスクレコーダの構成図である。この装置は、図1に示すように、バス100、ホストCPU101、RAM102、ROM103、ユーザインタフェース104、システムクロック105、光ディスク106、ピックアップ107、ECC(Error Correcting Coding)デコーダ108、ECCエンコーダ109、再生用バッファ110、記録/アフレコ用バッファ111、デマルチプレクサ112、マルチプレクサ113、多重化用バッファ114、オーディオデコーダ115、ビデオデコーダ116、オーディオエンコーダ117、ビデオエンコーダ118、および図示しないカメラ、マイク、スピーカ、ディスプレイ等で構成される。

【0016】ホストCPU101は、バス100を通じてデマルチプレクサ112、マルチプレクサ113、ピックアップ115、ビデオデコーダ116、オーディオエンコーダ117、ビデオエンコーダ118との通信を行う。

【0017】再生時に、光ディスク106からピックアップ107を通じて読み出されたデータは、ECCデコーダ108によって誤り訂正され、再生用バッファ110に一旦蓄えられる。デマルチプレクサ112はオーディオデコーダ115、ビデオデコーダ116からデータ送信要求に従い、再生用バッファ中のデータをその種類によって適当なデコーダに振り分ける。

【0018】一方、記録時に、オーディオエンコーダ117とビデオエンコーダ118によって圧縮符号化されたデータは、多重化用バッファ114に一旦送られ、マルチプレクサ113によってAV多重化され、記録/アフレコ用バッファ111に送られる。記録/アフレコ用バッファ111中のデータは、ECCエンコーダ109によって誤り訂正符号を付加され、ピックアップ107を通じて光ディスク106に記録される。

【0019】オーディオデータの符号化方式にはMPEG-1 Layer-IIを、ビデオデータの符号化方式にはMPEG-2をそれぞれ用いる。

【0020】光ディスク106は、外周から内周に向かって螺旋状に記録再生が行われる既習可能な光ディスクとする。2048byteを1セクタとし、誤り訂正のため16セクタでECCブロックを構成する。ECCブロック中のデータを

書き換える場合、そのデータが含まれるECCブロック全体を読み込み、誤り訂正を行って、対象のデータを書き換え、再び誤り訂正符号を付加し、ECCブロックを構成して、記録媒体に記録する必要がある。また、光ディスク106は、記録効率を上げるためCIN(回転角速度一定)を採用しており、記録側は回転数の異なる複数のゾーンで構成される。

【0021】<ファイルシステム>光ディスク106上の各種情報を管理するためにファイルシステムを用いる。ファイルシステムには、パーソナルコンピュータ(PC)との相互運用を考慮してUDF(Universal Disk Format)を使用する。ファイルシステム上では、各種管理情報やAVストリームはファイルとして扱われる。

【0022】ユーザエリアは、2048byteの論理ブロック(セクタと一対一対応)で管理される。各ファイルは、整数個のエクステンツ(連続した論理ブロック)で構成され、エクステンツ単位で分散して記録しても良い。空き領域は、Space Bitmapを用いて論理ブロック単位で管理される。

【0023】<ファイルフォーマット>AVストリーム管理のためのフォーマットとして、QuickTimeファイルフォーマットを用いる。QuickTimeファイルフォーマットとは、Apple社が開発したマルチメディアデータ管理用フォーマットであり、PCの世界で広く用いられている。

【0024】QuickTimeファイルフォーマットは、ビデオデータやオーディオデータ等(これらを総称してメディアデータとも呼ぶ)と管理情報とで構成される。両者を合わせてここでは、QuickTimeムービー(略してムービー)と呼ぶ。両者は同じファイル中に存在しても、別々のファイルに存在しても良い。

【0025】同じファイル中に存在する場合は、図2(a)に示すような構成をとる。各種情報はatomという共通の構造に格納される。管理情報はMovie atomという構造に格納され、AVストリームはMovie data atomという構造に格納される。尚、Movie atom中の管理情報には、メディアデータ中の任意の時間に対応するAVデータのファイル中の相対位置を導くためのテーブルや、メディアデータの属性情報や、後述する外部参照情報が含まれている。

【0026】一方、管理情報とメディアデータを別々のファイルに格納した場合は、図2(b)に示すような構成をとる。管理情報はMovie atomという構造に格納され、AVストリームはatomに格納される必要はない。

このとき、Movie atomはAVストリームを格納したファイル(「外部参照」している、という。

【0027】外部参照は、図2(c)に示すように、複数のAVストリームファイルに対して行うことが可能であり、この仕組みにより、AVストリーム自体を物理的に移動することなく、見かけ上編集を行ったように見える。いわゆる「ノンリニア編集」「非破壊編集」が可能にな

る。

【0028】それでは、図3乃至図12を用いて、QuickTimeの管理情報のフォーマットについて説明する。まず、共通の情報格納フォーマットであるatomについて説明する。atomの先頭には、そのatomのサイズであるAtom size、そのatomの種類情報であるTypeが必要存在する。Typeは4文字で区別され、例えばMovie atomでは'mov'、Movie data atomでは'mdat'となっている。

【0029】各atomは別のatomを含むことができる。すなわち、atom間には階層構造がある。Movie atomの構成を図3に示す。Movie header atomは、そのMovie atomを管理するムービーの全体的な属性を管理する。Track atomは、そのムービーに含まれるビデオやオーディオ等のトラックに関する情報を格納する。User data atomは、独自に定義可能なatomである。

【0030】Track atomの構成を図4に示す。Track header atomは、そのトラックの全体的な属性を管理する。Edit atomは、メディアデータのどの区間を、ムービーのどのタイミングで再生するかを管理する。Track reference atomは、別のトラックとの関係を管理する。Media atomは、実際のビデオやオーディオといったデータを管理する。User data atomは、メーカー独自の情報を管理する。本説明においては、このatomを含むトラックの種類、例えばオーディオトラック、サブオーディオトラック等に関する情報を格納する。

【0031】Track header atomの構成を図5に示す。ここでは、後での説明に必要なもののみについて説明する。flagsは属性を示すフラグの集合である。代表的なものとして、Track enabledフラグがあり、このフラグがであれば、そのトラックは再生され、0であれば再生されないと。layerはそのトラックの空間的な離散度を表しており、画像を表示するトラックが複数あれば、layerの値が小さいトラックほど画像が前面に表示される。また、Track IDは、このatomを内包するトラックのIDを示し、同じIDのトラックは個々のムービー内には複数存在しない。

【0032】Media atomの構成を図6に示す。Media header atomは、そのMedia atomの管理するメディアデータに関する全体的な属性等を管理する。Handler reference atomは、メディアデータをどのデコーダでデコードするかを示す情報を格納する。Media information atomは、ビデオやオーディオ等メディア固有の属性情報を管理する。

【0033】Media information atomの構成を図7に示す。Media information header atomは、ビデオやオーディオ等メディア固有の属性情報を管理する。Handler reference atomは、Media atomの項で説明した通りであり、Data information atomは、そのQuickTimeムービーが参照するメディアデータを包含するファイルの名前を管理するatomであるData reference atomを含む。Sample ta

ble atomは、データのサイズや再生時間等を管理している。

【0034】次に、Sample table atomについて説明するが、その前に、QuickTimeにおけるデータの管理方法について、図8を用いて説明する。QuickTimeでは、データの最小単位(例えばビデオフレーム)をサンプルと呼ぶ。個々のトラック毎に、サンプルには再生時間順に1から番号(サンプル番号)がついている。

【0035】また、QuickTimeフォーマットでは、個々のサンプルの再生時間長およびデータサイズを管理している。また、同一トラックに属するサンプルが再生時間順にファイル中で連続的に配置された領域をチャUNKと呼ぶ。チャUNKにも再生時間順に、1から番号がついている。

【0036】さらに、QuickTimeフォーマットでは、個々のチャUNKのファイル先頭からのアドレスおよび個々のチャUNKが含むサンプル数を管理している。これらの情報に基づき、任意の時間に対応するサンプルの位置を求めることが可能となっている。

【0037】Sample table atomの構成を図9に示す。Sample description atomは、個々のチャUNKのデータフォーマット(Data format)やサンプルが格納されているファイルのチャUNKのIndex等を管理する。Time-to-sample atomは、個々のサンプルの再生時間を管理する。

【0038】Sync sample atomは、個々のサンプルのうち、デコード開始可能なサンプルを管理する。Sample-to-chunk atomは、個々のチャUNKに含まれるサンプル数を管理する。Sample size atomは、個々のサンプルのサイズを管理する。Chunk offset atomは、個々のチャUNKのファイル先頭からのアドレスを管理する。

【0039】Edit atomは、図10に示すように、1個のEdit list atomを含む。Edit list atomはNumber of entriesで指定される個数分の、Track duration、Media time、Media rateはそのエントリを持つ。各エントリは、トラック上で連続的に再生される区間に対応し、そのトラック上での再生時間順に並んでいる。

【0040】Track durationはそのエントリが管理する区間のトラック上での再生時間、Media timeはそのエントリが管理する区間の先頭に対応するメディアデータ上の位置、Media rateはそのエントリが管理する区間の再生スピードを表す。尚、Media timeが1の場合は、そのエントリのTrack duration分、そのトラックでのサンプルの再生を停止する。この区間のことをempty editと呼ぶ。

【0041】図11にEdit listの使用例を示す。ここでは、Edit list atomの内容が図11(a)に示す内容であり、さらにサンプルの構成が図11(b)であったとする。尚、ここでは1番目のエントリのTrack durationをD(i)、Media timeをT(i)、Media rateをR(i)とす

る。このとき、実際のサンプルの再生は、図11(c)に示す順に行われる。このことについて簡単に説明する。

【0042】まず、エントリ#1はTrack durationが13000、Media timeが20000、Media rateが1であるため、そのトラックの先頭から13000の区間はサンプル中の時刻20000から33000の区間を再生する。次に、エントリ#2はTrack durationが5000、Media timeが1であるため、トラック中の時刻13000から18000の区間、何も再生を行わない。

【0043】最後に、エントリ#3はTrack durationが10000、Media timeが0、Media rateが1であるため、トラック中の時刻18000から28000の区間において、サンプル中の時刻0から10000の区間を再生する。

【0044】図12にTrack reference atomの構成を示す。このatomは、このatomを内包するトラックと他のトラックとの関係を管理している。トラック間の関係には、複数のタイプがあり、タイプ毎にTrack reference type atomを設ける。Track reference type atomの構成について説明する。Typeは、このatomが管理する関係のタイプ、例えばはタイプトラック間の同期を示す“sync”やチャプター区切りを示す“chapt”等を格納する。track-k-Idにはこのatomを内包するトラックと、Typeで示される関係にあるトラックのTrack IDを格納する。

【0045】図13にUser data atomの構成を示す。このatomは、QuickTimeフォーマットで定義されていない独自の情報は任意個数格納することができ、1個の独自の情報は1個のエントリで管理され、1個のエントリはSizeとTypeとUser dataで構成される。Sizeはそのエントリ自体のサイズを表し、Typeは独自情報をそれぞれ区別するための識別情報、User dataは実際のデータを表す。

【0046】AVストリームの形態>まず、本実施例におけるAVストリームの構成について、図14及び図15を用いて説明する。AVストリームは整数値のRecord Unit(RU)で構成される。RUはディスク上で連続的に記録する単位である。RUの長さは、AVストリームを構成するRUをどのようににディスク上に配置してもシーMLS再生(再生中に画像や音声が発切れないで再生できること)やリアルタイムアフレコ(アフレコ対象のビデオをシーMLS再生しながらオーディオを記録すること)が保証されるように設定される。この設定方法については後述する。

【0047】また、RU境界がEOPブロック境界と一致するようにストリームを構成する。RUのこれらの性質によって、AVストリームをディスクに記録した後も、シーMLS再生を保証したまま、ディスク上でRU単位の配置を容易に変更することができる。

【0048】RUは、整数個のVideo Unit(VU)で構成される。VUは単独再生可能な単位であり、そのことから再

生の際のエントリ・ポイントとなりうる。

【0049】VU構成を図15に示す。VUは、1秒程度のビデオデータを格納した整数個のGOP(グループ・オブ・ピクチャ)と、それらと同じ時間に再生されるメインオーディオデータを格納した整数個のAAU(オーディオ・アクセス・ユニット)とから構成される。

【0050】尚、GOPは、MPEG-2ビデオ規格における画像圧縮の単位であり、複数のビデオフレーム(厳密的には15フレーム程度)で構成される。AAUはMPEG-1 Layer-II規格における音声圧縮の単位で、1152点の音波形状サンプル点により構成される。サンプリング周波数が48kHzの場合、AAUあたりの再生時間は0.024秒となる。VU中では、AV同期再生のために必要となる遅延を小さくするため、AAU、GOPの順に配置する。

【0051】また、VU単位で独立再生を可能とするために、VU中のビデオデータの先頭にはSequence Header(SH)を置く。VUの再生時間は、VUに含まれるビデオフレーム数にビデオフレーム周期をかけたものと定義する。さらに、VUを整数個組み合わせてRUを構成する場合、RUの始終端をEOPブロック境界に合わせるため、VUの末尾を0で埋める。

【0052】AVストリーム管理方法>AVストリームの管理方法は、前述のQuickTimeフォーマットをベースにしている。図16にAVストリーム管理形態を示す。ビデオトラックは、各ビデオフレームを1サンプル(ビデオサンプル)、VU中のビデオの塊を1チャンク(ビデオチャンク)として管理する。メインオーディオトラックは、AAUを1サンプル(オーディオサンプル)、VU中のオーディオの塊を1チャンク(オーディオチャンク)として管理する。

【0053】<RU単位決定方法>次に、RU単位決定方法について説明する。この決定方法では、基準となるデバイス(Ri)のリアレンス・デバイス・モデルを想定し、その上でシーMLS再生が破綻しないように連続記録単位を決める。

【0054】それではまず、リアレンス・デバイス・モデルについて、図17を用いて説明する。リアレンス・デバイス・モデルは1個のピクチャップとそれにつながるEOPエンコンダ・デコーダ501、トラックパッファ502、デマルチプレクサ503、アフレコ用パッファ504、オーディオエンコーダ509、ビデオパッファ505、オーディオパッファ506、ビデオデコーダ507、オーディオデコーダ508とによって構成される。

【0055】本モデルにおけるシーMLS再生は、VUのデコード開始時にトラックパッファ502上に少なくとも1個VUが存在すれば保証されるものとする。オーディオフレームデータのEOPエンコンダ501へのデータの流入速度およびEOPデコーダ501からデータの出力速度はRsとする。

【0056】また、アクセスによる読み出し、記録の停

止する最大期間をTaとする。さらに、短いアクセス(100トラック程度)に要する時間をTkとする。なお、これら期間は、シーク時間、回転待ち時間、アクセス後に最初にディスクから読み出したデータがEOPから出力されるまでの時間が含まれる。本実施例では、R=20Mbps、T=0秒、Tk=0.2秒とする。

【0057】前記リアレンス・デバイス・モデルにおいて再生を行った場合、次のような条件を満たせば、トラックパッファ502のアンダーフローがないことが保証できる。

【0058】条件を示す前にまず、記号の定義を行う。AVストリームを構成する1番目の連続領域をG1とし、G1中に含まれる再生時間をTg(1)とする。Tg(1)はG1中に先頭が含まれているVUの再生時間の合計とする。また、G1からG1+i-1へのアクセス時間をlaとする。

【0059】また、再生時間Tg(i)分のVU読み出し時間をTr(i)とする。このとき、トラックパッファ502をアンダーフローさせない条件とは、分断ジャンプを含めた最大読み出し時間をTr(i)としたとき、任意のG1において、

$$Tg(i) \geq Tr(i) + la \dots \text{式1}$$

が成立することである。

【0060】なぜなら、この式は、シーMLS再生の十

分条件である、

$$Tg(i) \geq Tr(i) + la$$

【数1】

【0062】を満たす十分条件であるためである。

【0063】<式1>中のTr(i)に、Tr(i)=Tg(i)×(Rv/Ra) a/Rsを代入して、Tg(i)で解くとシーMLS再生を保証可能なTg(i)の条件

$$Tg(i) \geq (la \times Ra) / (Rs \times Rv - Ra) \dots \text{式2}$$

が得られる。

【0064】つまり、各連続領域に先頭の含まれるVUの合計が上式を満たすようにすれば、シーMLS再生を保証可能である。このとき、各連続領域には合計の再生時間が上式を満たす完全なVU群を含むように制限しても良い。

【0065】自動分割ムービーファイルでもく式2>を満たす必要がある。ただし、先頭の自動分割ムービーの最初のRUおよび末尾の自動分割ムービーの最後のRUはく式2>を満たさなくてもよい。なぜなら、先頭は記録媒体からのデータ読み出し開始より再生開始を遅らせることにより吸収でき、末尾については次に続くデータがないため連続再生を気にする必要が無いからである。このように先頭と末尾において条件を緩めることにより、短い空き領域を有効利用できる。

【0066】<インデックス・ファイル>ディレクトリに

種コンテンツ(以後、AVファイルと呼ぶ)を管理するため、AV Indexファイルという特別のQuickTimeムービーファイルをディスク内に1個置く。図18に、AV Indexファイルの構成を示す。AV Indexファイルは通常のQuickTimeムービーファイルと同様、管理情報であるMovie atom1791とデータ自体のMovie data atom1792で構成される。

【0067】AV Indexファイルは、複数のエントリを管理し、ディスク内の各AVファイルはそれぞれ1個のエントリで管理される。さらに、各AVファイルをまとめるための入れ物(以後フォルダと呼ぶ)等もそれぞれ1個のエントリで管理する。

【0068】Movie atom1791は、各エントリの属性情報を管理するためのProperty track1793、各エントリのタイトル文字列データを管理するためのTitle track1794、各エントリのサムネイル画像データを管理するためのThumbnail track1795、各エントリの代表オーディオデータを管理するためのIntro music track1796の計4種類のトラックで構成される。

【0069】各エントリに関する属性情報、それぞれは1792~1795のトラックのサンプルとして管理される。

例えばAVファイル1740に関する属性情報はProperty track1793上のサンプル1701、タイトル文字列データはTitle track1794上のサンプル1711、サムネイル画像データはThumbnail track1795上のサンプル1721、代表オーディオデータはIntro music track1796上のサンプル1731で管理する。

【0070】サンプル間の対応付けは、各サンプルの再生開始時間に基づき行う。すなわち、トラック間で同一時刻に位置するサンプルが同一エントリに対応していると判断する。

【0071】Movie data atom1793は、各AVファイルに関する属性情報や、タイトル文字列データ、サムネイル画像データ、代表オーディオデータを格納する。属性情報は図19に示す構成を取る。各フィールドについて説明する。Versionは、ファイルフォーマットのバージョンを示す。pe-flagsは各種フラグをまとめたものであり、詳細は後述する。

【0072】parent-entry-numberは、属性情報に対応するエントリが属するフォルダに対応するエントリのentry-numberを格納し、entry-numberは、属性情報に対応するエントリEntry-numberを格納する。この2個の情報で、ファイルとフォルダの包含関係を表す。set-depend-ent-flagsおよびuser-private-flagsについては、説明を省略する。

【0073】creation-timeおよびmodification-timeはこの属性情報に対応するエントリが作成された日時、修正された日時を表す。durationはこの属性情報に対応するエントリの再生時間を表す。binary-file-identifierは、この管理情報に対応するエントリがファイルに対応

している場合、そのファイルのパス名を固定長にエンコードしている場合、そのファイルが他のファイルから参照されている。詳細については説明は省略する。

【0074】referred-counterはこの属性情報に対応するエンタリが管理するファイルが他のファイルから参照されている回数格納する。referrer file listは、実際に参照しているファイルのパス名のリストを格納する。URL file identifierは、管理するファイルが上記のbinary-file-identifierにエンコードできない場合にURL (Unified Resource Locator) 形式で、ファイルのパスを格納する。

【0075】<記録時の処理>ユーザから録画が指示された場合の処理を、図20に示して説明する。このとき記録するAVストリームは、ビデオのビットレートR=5Mbps、オーディオのサンプリング周波数48kHz、ビットレートR=192000bpsであるものとする。すなわち、ファイルシステムの管理情報はRAM102上に読み込まれていると仮定する。

【0076】まず、ストリームの構成や連続領域の構成を決定する(ステップ701)。1VUを160P=30フレームで構成するとき、<式4>にR=20Mbps、T=1秒、R=5Mbps、R=256kbpsを代入し、Te(i)の範囲である1.36秒以上が得られる。1VUの再生時間を0.5秒としているため、R再生時間は2秒とする。

【0077】次に、ムービーファイル記録準備を行う。具体的にはファイルを開き、1個のRUを連続的に記録可能な空き領域を探す。存在しなければ録画を中止し、録画できないことをユーザに知らせる(ステップ702)。【0078】また、オーディオエンコーダ117、ビデオエンコーダ118をそれぞれ起動する(ステップ703)。次に、記録用バッファ111に1RU分のデータが蓄積されているかどうかチェックする(ステップ704)。

【0079】蓄積されれば、記録用バッファ111中の1RU分のデータを連続的に記録する(ステップ705)。蓄積されていないければ、記録終了が指示されていないかどうかチェックし(ステップ706)、指示されていないければステップ704を再行し、指示されれば以下の記録終了処理を行う。

【0080】まず、現在記録中のムービーファイルに残りデータ(ステップ707)および管理情報を記録する(ステップ708)。最後に、AV Indexファイルに今回作成したQuickTimeムービーファイルを登録する(ステップ709)。なお、ビデオトラックとオーディオトラックのトラック種別は「オリジナル」に設定する。

【0081】<サブオーディオ付加時の処理>既に録画したビデオに対して、オーディオ(サブオーディオ)データを付加する際の処理について、図21乃至図24を用いて説明する。ユーザからのサブオーディオデータ付加方法として次の2種類が考えられる。

(1) 新規に入力したオーディオデータを付加(アプレ

コ) (2) 既記録のオーディオデータを指定して付加 それぞれについて以下で説明を行う。

【0082】<サブオーディオ付加時の処理、新規入力オーディオを付加>まず、新規に入力したオーディオデータを付加する。いわゆるアフレコを行う場合について説明する。図2に示すように、光ディスク106上には、ビデオデータとオーディオデータの格納されたQuickTimeムービーファイル202と前記QuickTimeムービーファイルの登録されたAV Indexファイル201が記録されているものとする。

【0083】QuickTimeムービーファイル202に格納されているビデオデータおよびオーディオデータは、それぞれビデオトラック201とメインオーディオトラック202で管理されているものとする。このときQuickTimeムービーファイル202に対して、新規に入力するオーディオデータをサブオーディオとして付加する場合の処理を、図22に示すフローチャートを用いて説明する。

【0084】まず、初期状態として再生可能なQuickTimeムービーファイルのリストをGUI画面上に表示する(ステップ801)。このリストはAV Indexファイル201に記録されている情報を元に作成されたものである。ユーザはムービーファイル202にサブオーディオを付加したいため、前記リストからQuickTimeムービーファイル202を選択し、再生を指示し、その結果、ムービーファイル202の再生を開始する(ステップ802)。

【0085】次に、ユーザはアフレコを開始したい箇所で一時的停止を指示し、再生を一時的に停止する(ステップ803)。次に、ユーザはアフレコ開始を指示し、その結果アフレコを開始する(ステップ804)。そのとき、アフレコを開始する(ステップ804)。そのとき、アフレコを開始する(ステップ804)。そのとき、アフレコを開始する(ステップ804)。

【0086】アフレコ中はオーディオエンコーダ117から入力されるオーディオデータをQuickTimeムービーファイル203に記録する(ステップ805)。ユーザからアフレコ停止が指示されたら、入力されたオーディオデータに関する管理情報をMovie atomの形でファイル203に格納する(ステップ806)。次に、追加するサブオーディオを管理するトラック2023をQuickTimeムービーファイル2002に作成する(ステップ807)。

【0087】トラック2023のトラック種別は「サブオーディオ」に設定する。その直後に、ユーザに対して「ビデオに従属(ビデオと同期して編集される)」「ビデオと独立(ビデオの編集と無関係)」の選択画面を表示する(ステップ808)。新規入力の場合、ビデオに従属したものである可能性が高いため、初期値は「ビデオに従属」に設定しておく。

【0088】ユーザが「ビデオに従属」を選択した場合、まず、トラック2023のTrack header atom中にTrack

reference atomにtypeが"sync"のTrack reference type atomを追加し、そのTrack reference type atomのTrack DIsにQuick Timeムービーファイル202のビデオトラック202のTrack IDを格納する(ステップ810)。

【0089】次に、ユーザに対して、入力されたオーディオデータを、別のコンテンツでも再利用する可能性があるかどうかを確認し(ステップ811)、可能性のある場合、ムービーファイル2023をAV Indexファイル2001に登録する(ステップ812)。

【0090】次に、そのトラック2023に対し、ムービーファイル2003の情報を元にSample table atomを追加する。このとき、データの参照先はムービーファイル2003とする。さらに、そのトラックに対し、サブオーディオの再生開始点、つまり現在一時停止している箇所を元にEdit list atomを作成する(ステップ813)。

【0091】最後に、RAM102にあるQuickTimeムービーファイル202の管理情報を光ディスク106上のQuickTimeムービーファイル202のMovie atomに反映させ(ステップ814)、光ディスク106上のAV Indexファイル2001中のQuickTimeムービーファイル202に対応するエントリのReferred-countを増加させる(ステップ815)。

【0092】<サブオーディオ付加時の処理：既記録オーディオを付加>次に、既記録のオーディオデータを指定して付加する場合について説明する。図24に示すように、光ディスク106上には、ビデオデータとオーディオデータの格納されたQuickTimeムービーファイル2102と、オーディオデータのみに格納したQuickTimeムービーファイル2103と、前記QuickTimeムービーファイルの登録されたAV Indexファイル2101が記録されているものとする。

【0093】QuickTimeムービーファイル2102に格納されているビデオデータおよびオーディオデータは、それぞれビデオトラック2121とメインオーディオトラック2112で管理されているものとする。また、QuickTimeムービーファイル2103に格納されているオーディオデータは、それぞれオーディオトラック2122で管理されているものとする。

【0094】このときQuickTimeムービーファイル2102にQuickTimeムービーファイル2103をサブオーディオとして付加する場合の処理を、図25に示すフローチャートを用いて説明する。

【0095】まず、初期状態として再生可能なQuickTimeムービーファイルのリストをGUI画面上に表示する(ステップ901)。このリストはAV Indexファイル2101に記録されている情報を元に作成されたものである。ユーザはムービーファイル2102にサブオーディオデータを付加したいため、前記リストからQuickTimeムービーファイル2102を選択し、再生を指示する。その結果、ムービーファイル2102の再生を開始する(ステップ902)。

【0096】次に、ユーザはサブオーディオを付加した

い箇所で一時的停止を指示し、再生を一時的に停止する(ステップ903)。次に、サブオーディオ付加を指示し、その指示に応じて、AV Indexファイル2101に登録されているオーディオデータのみのQuickTimeムービーファイルのリストをGUI画面に表示する(ステップ904)。このリストの中からユーザはムービーファイル2103を選択する。【0097】次に、追加するサブオーディオを管理するトラック2123をQuickTimeムービーファイル2102に作成する(ステップ905)。トラック2123のトラック種別は「サブオーディオ」に設定する。その直後に、ユーザに対して「ビデオに従属(ビデオと同期して編集される)」「ビデオと独立(ビデオの編集と無関係)」の選択画面を表示する(ステップ906)。なお、ファイルから選択した場合は前記のアフレコと比べ同期させない可能性が高いため、初期値は「ビデオと独立」に設定しておく。

【0098】ユーザが「ビデオに従属」を選択した場合、トラック2123のTrack header atom中にTrack reference atomにtypeが"sync"のTrack reference type atomを追加し、そのTrack reference type atomのTrack DIsにQuick Timeムービーファイル2102のビデオトラック2121のTrack IDを格納する(ステップ908)。

【0099】次に、トラック2123に対し、QuickTimeムービーファイル2103の情報を元にSample table atomを追加する。このとき、データの参照先はムービーファイル2103とする。さらに、そのトラックに対し、サブオーディオの再生開始点、つまり現在一時停止している箇所を元にEdit list atomを作成する(ステップ909)。

【0100】最後に、RAM102上にあるQuickTimeムービーファイル202の管理情報を光ディスク106上のQuickTimeムービーファイル202のMovie atomに反映させ(ステップ910)、光ディスク106上のAV Indexファイル2101中のQuickTimeムービーファイル2102に対応するエントリのReferred-countを増加させる(ステップ911)。

【0101】<削除処理>本実施形態における、部分削除時の処理について、図26を例にとって説明する。ここでは、ユーザから図26に示すようにQuickTimeムービーファイル2002の部分範囲2031〜2032の削除が指示されたとする。このときの処理を、図27に示すフローチャートを用いて説明する。

【0102】まず、削除対象のQuickTimeムービーファイルにサブオーディオトラックが存在するかどうかチェックする(ステップ1001)。存在しなければステップ1004を実行する。存在したなら、そのサブオーディオトラック(トラック2023)がビデオトラック(トラック2021)と同期しているかどうかを調べ(ステップ1002)。【0103】具体的にはTrack header atom中のTrack reference atomにtypeが"sync"のTrack reference type atomが存在し、track-IDIsにビデオトラックのTrack IDが含まれていれば、同期していると判断する。同期し

ていなければステップ1004を実行し、同期していれば、部分区間T2031〜T2032に対応するサブオーディオトラックの区間(区間2053)が再生対象から除外される(ステップ1003)。

【0104】次に、ビデオトラックおよびメインオーディオトラック中の部分区間T2031〜T2032に対応する区間(区間2051および2052)が再生対象から除外される(ステップ1004)。次に、上記の編集結果を光ディスク106上のAV7ファイル(QuickTimeムービーファイル2002)のMovie atomに反映させる(ステップ1005)。

【0105】最後に、上記の編集結果を光ディスク106上のAV Indexファイル(AV Indexファイル2001)に反映させる。具体的には、編集に伴うDurationやModification-time等の変化を反映させる。

【0106】以上の処理の結果、サブオーディオトラック2023がビデオトラック2021と同期していた場合、図28に示すように、ビデオトラック2021、メインオーディオトラック2022、サブオーディオトラック2023について、それぞれ区間2051、2052、2053が再生対象から除外され、サブオーディオトラック2023は削除前と変わりがなく、図28の場合、ビデオトラック2021の区間2061以降の区間は、サブオーディオトラック2023の再生されることになり、不自然になるため、ビデオトラック2021の区間2061以降の区間に対応するサブオーディオトラック2023の区間を再生対象から外してもよい。

【0108】つまり、ユーザはサブオーディオを登録する際に一度属性を指定するだけで、編集時に特に指定することなく、自分の意図した編集(ビデオを部分削除することによって、BGMが不連続になったり、吹き替え音声だけが残ったりすることがない)が可能となる。

【0109】ここでは説明は省略するが、T2011にビデオを挿入する場合も上記の説明と同様、サブオーディオトラック2023がビデオトラック2021と同期している場合、サブオーディオトラック2023の区間T2031の区間に再生区間を挿入し、同期しない場合、サブオーディオトラック2023を変更しないの言うまでもない。

【0110】また、QuickTimeムービーファイル2002の削除がユーザから指示された場合、サブオーディオトラック2023から参照されているQuickTimeムービーファイル2003がAV Indexファイル2001に登録されていないければ、QuickTimeムービーファイル2003は再利用可能な状態ではないため、QuickTimeムービーファイル2002の削除と共にQuickTimeムービーファイル2003も削除する。

QuickTimeムービーファイル2003がAV Indexファイル2001に登録されている場合、QuickTimeムービーファイル2003は再利用可能な状態ではないため、QuickTimeムービーファイル2002の削除と共にQuickTimeムービーファイル2003も削除する。

QuickTimeムービーファイル2003がAV Indexファイル2001

50

に登録されていれば、QuickTimeムービーファイル2002のみ削除する。

【0111】さらに、本実施形態では、部分削除を、QuickTimeムービーファイル2002のMovie atom、すなわち管理情報の書き換えのみで実現したが、QuickTimeムービーファイル2002、2003に含まれる実際のAVデータを削除することが可能ないとは言ってもよい。

【0112】<バリエーション>本実施形態では、サブオーディオトラックとビデオトラックとの関係をTrack reference atomの情報で管理しているが、両トラックの関係を示せるものであれば、どのような形態であってもよいことは言うまでもない。例えば、AV Indexファイルのような、別ファイルで管理されているとも構わない。

【0113】また、本実施形態においては、サブオーディオ付加時に、ユーザに「ビデオと独立」「ビデオに依存」を選択させているが、システムが自動設定しても構わない。

【0114】さらに、本実施形態では、サブオーディオを取り上げているが、本発明はサブオーディオにのみ限られる。例えば、ビデオに重量表示するアニメーションに対して適用可能であることは言うまでもない。

【0115】そしてまた、本実施形態においては、サブオーディオデータを参照元のムービーファイルとは別のファイルに格納しているが、同一のファイルに格納しても構わない。

【0116】また、本実施形態においては、外部参照するオーディオデータファイルを、QuickTimeムービーファイルフォーマットで記録しているが、それ以外のフォーマット、たとえばMP3(MPEG 1 Audio Layer 3)フォーマットやM4Vフォーマット(Windows(登録商標)における標準オーディオフォーマット)であってもよい。ただし、サブオーディオトラックにおいてそれらのオーディオデータファイルを参照する際には、Sample table atomを構築するために、それらのファイルを解析する必要がある。

【0117】

【発明の効果】以上説明したように、本発明によれば、ビデオデータに対してサブオーディオデータを後から付加した際に、ビデオとの同期関係を指定することにより、削除、追加等の編集時にユーザは特に指定することなく、自分の意図した編集が可能となる。

【面の簡単な説明】

【図1】本発明の実施形態における概略構成を示すブロック図である。

【図2】QuickTimeファイルフォーマットにおける管理情報とAVストリームとの関係を示す説明図である。

【図3】QuickTimeファイルフォーマットにおけるMovie atomの概要を示す説明図である。

【図4】QuickTimeファイルフォーマットにおけるTrack atomの概要を示す説明図である。

50

【図5】QuickTimeファイルフォーマットにおけるTrack header atomの構成を示す説明図である。

【図6】QuickTimeファイルフォーマットにおけるMedia atomの構成を示す説明図である。

【図7】QuickTimeファイルフォーマットにおけるMedia information atomの構成を示す説明図である。

【図8】Sample table atomによるデータ管理の例を示す説明図である。

【図9】QuickTimeファイルフォーマットにおけるSample table atomの構成を示す説明図である。

【図10】QuickTimeファイルフォーマットにおけるEdit atomの構成を示す説明図である。

【図11】Edit atomによる再生範囲指定の例を示す説明図である。

【図12】QuickTimeファイルフォーマットにおけるTrack referenceの構成を示す説明図である。

【図13】QuickTimeファイルフォーマットにおけるUser data atomの構成を示す説明図である。

【図14】本発明におけるAVストリームの構成を示す説明図である。

【図15】本発明におけるVUの構成を示す説明図である。

【図16】本発明におけるAVストリーム管理形態を示す説明図である。

【図17】本発明におけるリファレンス・デバイス・モデルを示す説明図である。

【図18】本発明におけるAV Indexの構成を示す説明図である。

【図19】本発明におけるAV Index中の属性情報の構成を示す説明図である。

【図20】本発明における、録画動作を示すフローチャートである。

【図21】本発明における、サブオーディオデータ付加直前の状態の例を示す説明図である。

【図22】本発明における、新規入力サブオーディオ付加動作を示すフローチャートである。

【図23】本発明における、サブオーディオデータ付加直後の状態の例を示す説明図である。

【図24】本発明における、既記録データをサブオーディ

【図3】

Movie atom {  
Atom size  
Type('mov')  
Movie header atom  
Track atom (video track)  
Track atom (main audio track)  
User data atom  
}

【図4】

Track atom {  
Atom size  
Type('trak')  
Track header atom  
Edit atom  
Track reference atom  
Media atom  
User data atom  
}

【図6】

Media atom {  
Atom size  
Type('media')  
Media header atom  
Handler reference atom  
Media information atom  
User data atom  
}

【図5】

QuickTime atom {  
Atom size  
Type('qt')  
QuickTime header atom  
Media atom  
User data atom  
}

【図6】

QuickTime atom {  
Atom size  
Type('qt')  
QuickTime header atom  
Media atom  
User data atom  
}

【図5】

QuickTime atom {  
Atom size  
Type('qt')  
QuickTime header atom  
Media atom  
User data atom  
}

【図5】

QuickTime atom {  
Atom size  
Type('qt')  
QuickTime header atom  
Media atom  
User data atom  
}

【図5】

QuickTime atom {  
Atom size  
Type('qt')  
QuickTime header atom  
Media atom  
User data atom  
}

【図5】

QuickTime atom {  
Atom size  
Type('qt')  
QuickTime header atom  
Media atom  
User data atom  
}

【図5】

QuickTime atom {  
Atom size  
Type('qt')  
QuickTime header atom  
Media atom  
User data atom  
}

【図5】

QuickTime atom {  
Atom size  
Type('qt')  
QuickTime header atom  
Media atom  
User data atom  
}

【図5】

QuickTime atom {  
Atom size  
Type('qt')  
QuickTime header atom  
Media atom  
User data atom  
}

【図5】

QuickTime atom {  
Atom size  
Type('qt')  
QuickTime header atom  
Media atom  
User data atom  
}

【図5】

QuickTime atom {  
Atom size  
Type('qt')  
QuickTime header atom  
Media atom  
User data atom  
}

【図5】

QuickTime atom {  
Atom size  
Type('qt')  
QuickTime header atom  
Media atom  
User data atom  
}

【図5】

QuickTime atom {  
Atom size  
Type('qt')  
QuickTime header atom  
Media atom  
User data atom  
}

【図5】

QuickTime atom {  
Atom size  
Type('qt')  
QuickTime header atom  
Media atom  
User data atom  
}

【図5】

QuickTime atom {  
Atom size  
Type('qt')  
QuickTime header atom  
Media atom  
User data atom  
}

【図5】

QuickTime atom {  
Atom size  
Type('qt')  
QuickTime header atom  
Media atom  
User data atom  
}

【図5】

QuickTime atom {  
Atom size  
Type('qt')  
QuickTime header atom  
Media atom  
User data atom  
}

【図5】

QuickTime atom {  
Atom size  
Type('qt')  
QuickTime header atom  
Media atom  
User data atom  
}

【図5】

QuickTime atom {  
Atom size  
Type('qt')  
QuickTime header atom  
Media atom  
User data atom  
}

【図5】

QuickTime atom {  
Atom size  
Type('qt')  
QuickTime header atom  
Media atom  
User data atom  
}

【図5】

QuickTime atom {  
Atom size  
Type('qt')  
QuickTime header atom  
Media atom  
User data atom  
}

【図5】

QuickTime atom {  
Atom size  
Type('qt')  
QuickTime header atom  
Media atom  
User data atom  
}

【図5】

QuickTime atom {  
Atom size  
Type('qt')  
QuickTime header atom  
Media atom  
User data atom  
}

【図5】

QuickTime atom {  
Atom size  
Type('qt')  
QuickTime header atom  
Media atom  
User data atom  
}

【図5】

QuickTime atom {  
Atom size  
Type('qt')  
QuickTime header atom  
Media atom  
User data atom  
}

【図5】

QuickTime atom {  
Atom size  
Type('qt')  
QuickTime header atom  
Media atom  
User data atom  
}

【図5】

QuickTime atom {  
Atom size  
Type('qt')  
QuickTime header atom  
Media atom  
User data atom  
}

【図5】

QuickTime atom {  
Atom size  
Type('qt')  
QuickTime header atom  
Media atom  
User data atom  
}

【図5】

QuickTime atom {  
Atom size  
Type('qt')  
QuickTime header atom  
Media atom  
User data atom  
}

【図5】

QuickTime atom {  
Atom size  
Type('qt')  
QuickTime header atom  
Media atom  
User data atom  
}

【図5】

QuickTime atom {  
Atom size  
Type('qt')  
QuickTime header atom  
Media atom  
User data atom  
}

【図5】

QuickTime atom {  
Atom size  
Type('qt')  
QuickTime header atom  
Media atom  
User data atom  
}

【図5】

QuickTime atom {  
Atom size  
Type('qt')  
QuickTime header atom  
Media atom  
User data atom  
}

【図5】

QuickTime atom {  
Atom size  
Type('qt')  
QuickTime header atom  
Media atom  
User data atom  
}

【図5】

QuickTime atom {  
Atom size  
Type('qt')  
QuickTime header atom  
Media atom  
User data atom  
}

【図5】

QuickTime atom {  
Atom size  
Type('qt')  
QuickTime header atom  
Media atom  
User data atom  
}

【図5】

QuickTime atom {  
Atom size  
Type('qt')  
QuickTime header atom  
Media atom  
User data atom  
}

【図5】

QuickTime atom {  
Atom size  
Type('qt')  
QuickTime header atom  
Media atom  
User data atom  
}

【図5】

QuickTime atom {  
Atom size  
Type('qt')  
QuickTime header atom  
Media atom  
User data atom  
}

【図5】

QuickTime atom {  
Atom size  
Type('qt')  
QuickTime header atom  
Media atom  
User data atom  
}

【図5】

QuickTime atom {  
Atom size  
Type('qt')  
QuickTime header atom  
Media atom  
User data atom  
}

【図5】

QuickTime atom {  
Atom size  
Type('qt')  
QuickTime header atom  
Media atom  
User data atom  
}

【図5】

QuickTime atom {  
Atom size  
Type('qt')  
QuickTime header atom  
Media atom  
User data atom  
}

【図5】

QuickTime atom {  
Atom size  
Type('qt')  
QuickTime header atom  
Media atom  
User data atom  
}

【図5】

QuickTime atom {  
Atom size  
Type('qt')  
QuickTime header atom  
Media atom  
User data atom  
}

【図5】

QuickTime atom {  
Atom size  
Type('qt')  
QuickTime header atom  
Media atom  
User data atom  
}

【図5】

QuickTime atom {  
Atom size  
Type('qt')  
QuickTime header atom  
Media atom  
User data atom  
}

【図5】

QuickTime atom {  
Atom size  
Type('qt')  
QuickTime header atom  
Media atom  
User data atom  
}

【図5】

QuickTime atom {  
Atom size  
Type('qt')  
QuickTime header atom  
Media atom  
User data atom  
}

【図5】

QuickTime atom {  
Atom size  
Type('qt')  
QuickTime header atom  
Media atom  
User data atom  
}

【図5】

QuickTime atom {  
Atom size  
Type('qt')  
QuickTime header atom  
Media atom  
User data atom  
}

【図5】

QuickTime atom {  
Atom size  
Type('qt')  
QuickTime header atom  
Media atom  
User data atom  
}

【図5】

QuickTime atom {  
Atom size  
Type('qt')  
QuickTime header atom  
Media atom  
User data atom  
}

【図5】

QuickTime atom {  
Atom size  
Type('qt')  
QuickTime header atom  
Media atom  
User data atom  
}

【図5】

QuickTime atom {  
Atom size  
Type('qt')  
QuickTime header atom  
Media atom  
User data atom  
}

【図5】

QuickTime atom {  
Atom size  
Type('qt')  
QuickTime header atom  
Media atom  
User data atom  
}

【図5】

QuickTime atom {  
Atom size  
Type('qt')  
QuickTime header atom  
Media atom  
User data atom  
}

【図5】

QuickTime atom {  
Atom size  
Type('qt')  
QuickTime header atom  
Media atom  
User data atom  
}

【図5】

QuickTime atom {  
Atom size  
Type('qt')  
QuickTime header atom  
Media atom  
User data atom  
}

【図5】

QuickTime atom {  
Atom size  
Type('qt')  
QuickTime header atom  
Media atom  
User data atom  
}

【図5】

QuickTime atom {  
Atom size  
Type('qt')  
QuickTime header atom  
Media atom  
User data atom  
}

【図5】

QuickTime atom {  
Atom size  
Type('qt')  
QuickTime header atom  
Media atom  
User data atom  
}

【図5】

QuickTime atom {  
Atom size  
Type('qt')  
QuickTime header atom  
Media atom  
User data atom  
}

【図5】

QuickTime atom {  
Atom size  
Type('qt')  
QuickTime header atom  
Media atom  
User data atom  
}

【図5】

QuickTime atom {  
Atom size  
Type('qt')  
QuickTime header atom  
Media atom  
User data atom  
}

【図5】

QuickTime atom {  
Atom size  
Type('qt')  
QuickTime header atom  
Media atom  
User data atom  
}

【図5】

QuickTime atom {  
Atom size  
Type('qt')  
QuickTime header atom  
Media atom  
User data atom  
}

【図5】

QuickTime atom {  
Atom size  
Type('qt')  
QuickTime header atom  
Media atom  
User data atom  
}

【図5】

QuickTime atom {  
Atom size  
Type('qt')  
QuickTime header atom  
Media atom  
User data atom  
}

【図5】

QuickTime atom {  
Atom size  
Type('qt')  
QuickTime header atom  
Media atom  
User data atom  
}

【図5】

QuickTime atom {  
Atom size  
Type('qt')  
QuickTime header atom  
Media atom  
User data atom  
}

【図5】

QuickTime atom {  
Atom size  
Type('qt')  
QuickTime header atom  
Media atom  
User data atom  
}

【図5】

QuickTime atom {  
Atom size  
Type('qt')  
QuickTime header atom  
Media atom  
User data atom  
}

【図5】

QuickTime atom {  
Atom size  
Type('qt')  
QuickTime header atom  
Media atom  
User data atom  
}

【図5】

QuickTime atom {  
Atom size  
Type('qt')  
QuickTime header atom  
Media atom  
User data atom  
}

【図5】

QuickTime atom {  
Atom size  
Type('qt')  
QuickTime header atom  
Media atom  
User data atom  
}

【図5】

QuickTime atom {  
Atom size  
Type('qt')  
QuickTime header atom  
Media atom  
User data atom  
}

【図5】

QuickTime atom {  
Atom size  
Type('qt')  
QuickTime header atom  
Media atom  
User data atom  
}

【図5】

QuickTime atom {  
Atom size  
Type('qt')  
QuickTime header atom  
Media atom  
User data atom  
}

【図5】

QuickTime atom {  
Atom size  
Type('qt')  
QuickTime header atom  
Media atom  
User data atom  
}

【図5】

QuickTime atom {  
Atom size  
Type('qt')  
QuickTime header atom  
Media atom  
User data atom  
}

【図5】

QuickTime atom {  
Atom size  
Type('qt')  
QuickTime header atom  
Media atom  
User data atom  
}

【図5】

QuickTime atom {  
Atom size  
Type('qt')  
QuickTime header atom  
Media atom  
User data atom  
}

【図5】

QuickTime atom {  
Atom size  
Type('qt')  
QuickTime header atom  
Media atom  
User data atom  
}

【図5】

QuickTime atom {  
Atom size  
Type('qt')  
QuickTime header atom  
Media atom  
User data atom  
}

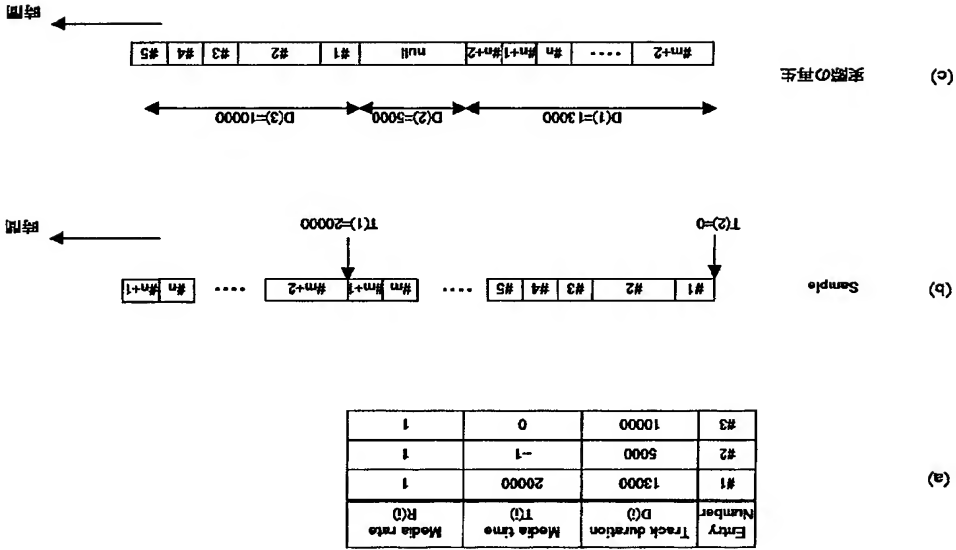
【図5】

Quick

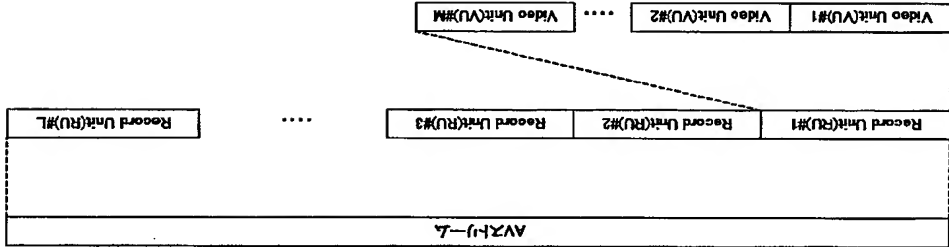




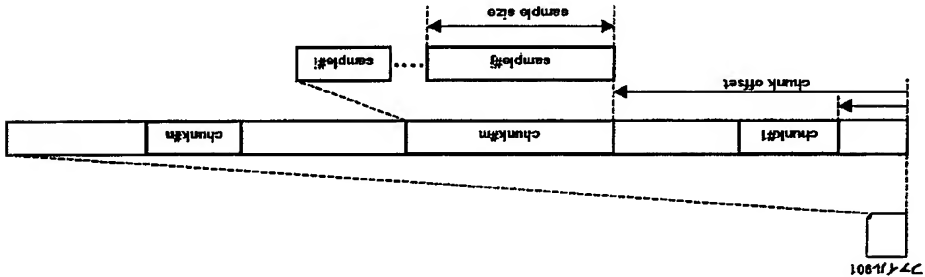
【図11】



【図14】

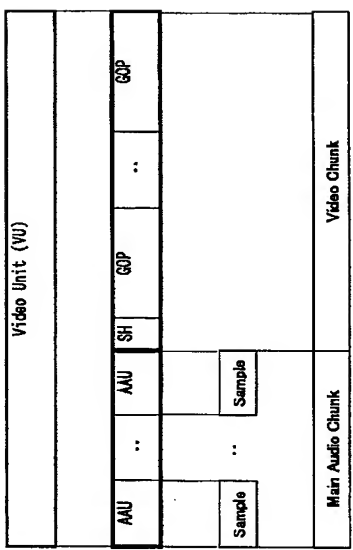


【図9】



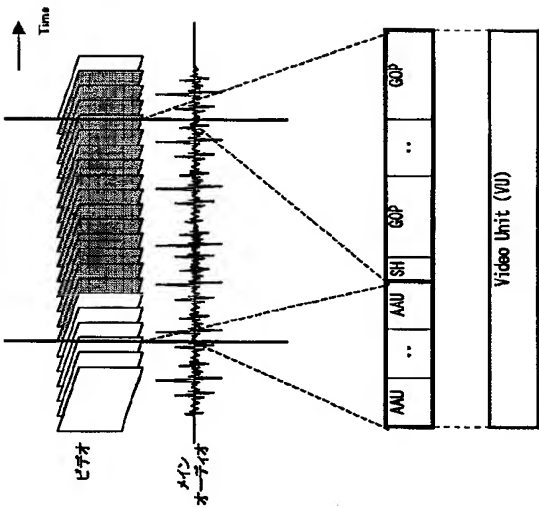


【図16】



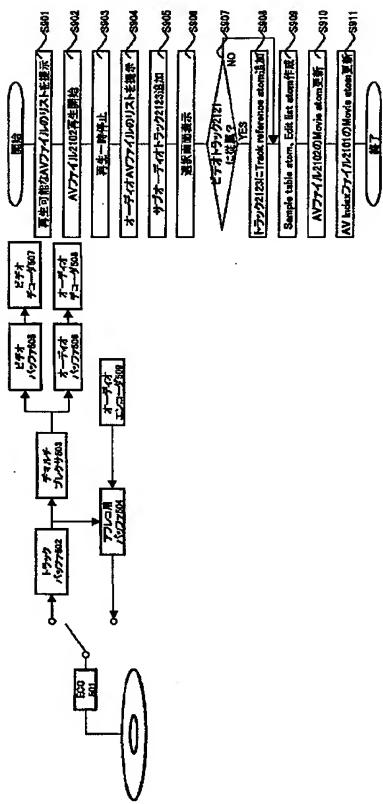
AAU: Audio Access Unit  
GOP: Group Of Pictures  
SH: Sequence Header

【図15】

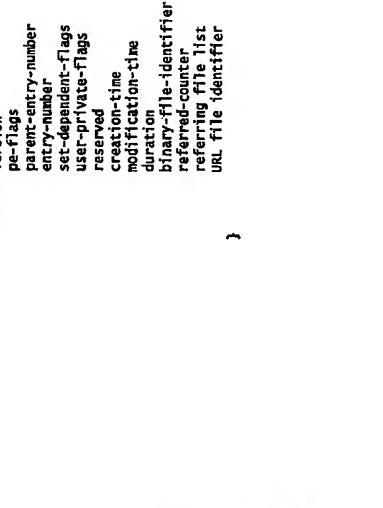


AAU: Audio Access Unit  
SH: Sequence Header

【図17】



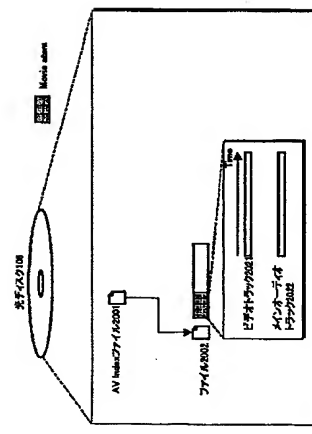
【図25】



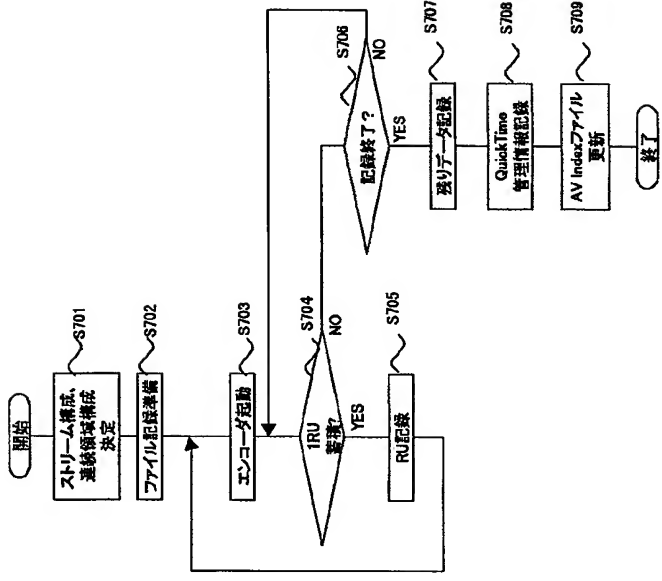
【図19】

Property Entry {  
version  
parent-entry-number  
entry-number  
set-dependent-flags  
user-private-flags  
reserved  
creation-time  
modification-time  
duration  
binary-file-identifier  
referenced-counter  
referring file list  
URL file identifier  
}

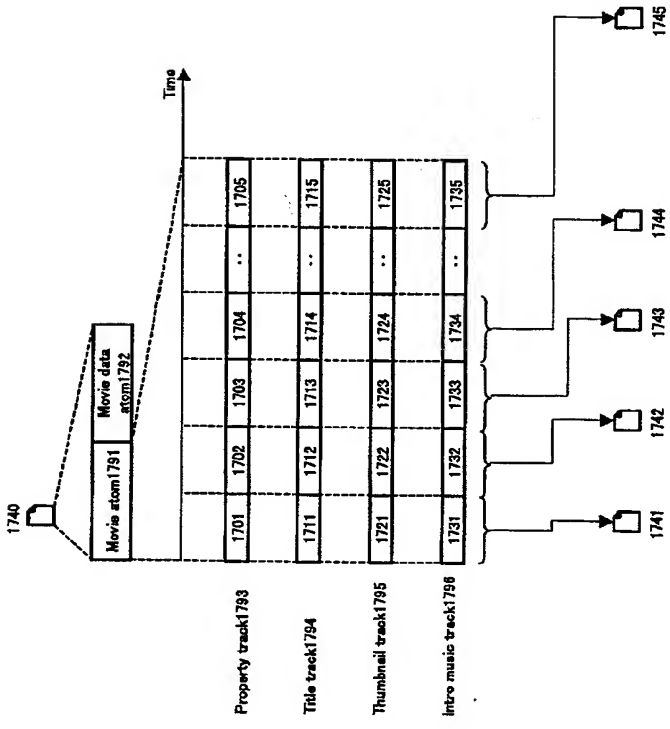
【図21】



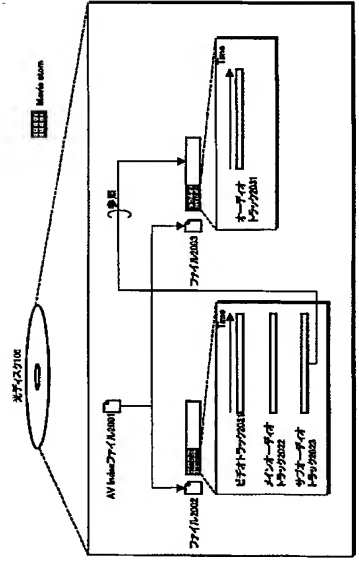
【図20】



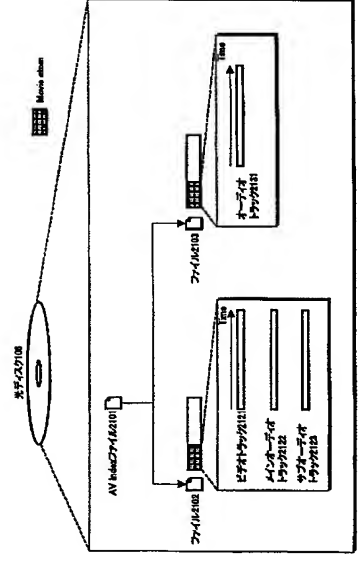
【図18】



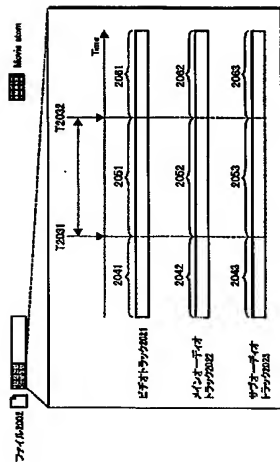
【図23】



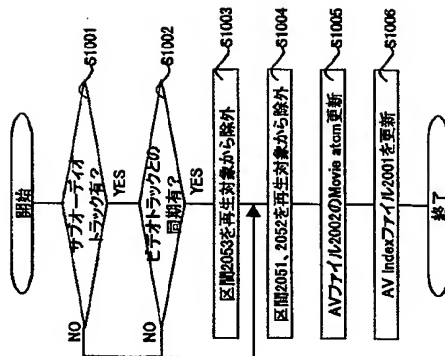
【図24】



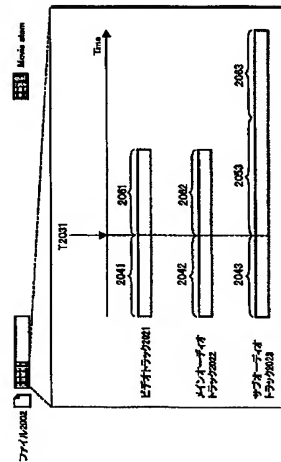
【図26】



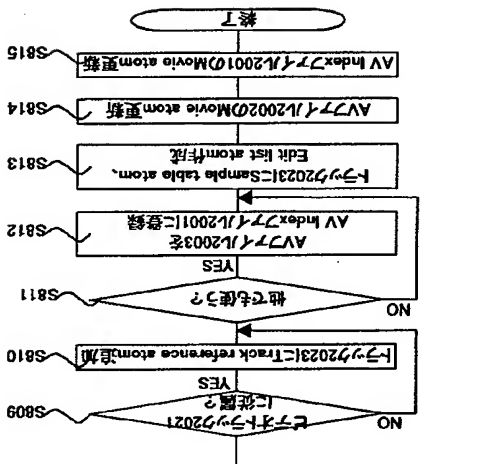
【図27】



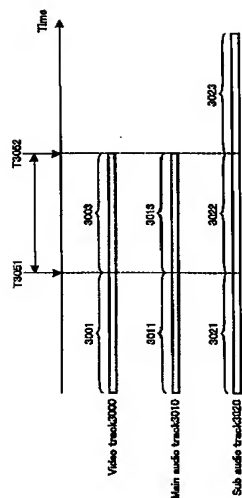
【図28】



【図22】



【図32】

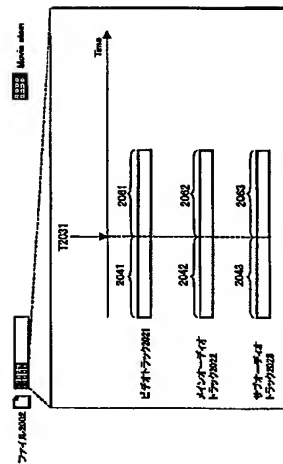


フロントページの続き

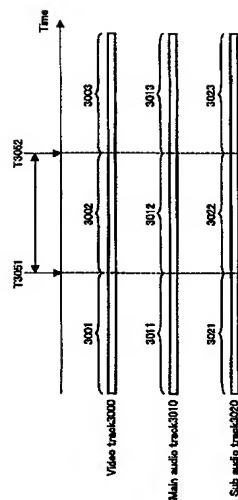
(72)発明者 山口 孝好  
大阪府大阪市阿倍野区長池町22番22号 シ  
ヤープ株式会社内

Fターム(参考) 5C053 FA23 GA11 GB15 GB37 HA30  
JA07 JA22 JA30  
5D044 AB07 BC04 CC06 DE32 DE48  
GK12 GM21 HL16  
5D110 AA14 AA29 BB01 BB20 CA05  
CA06 CA31 CB08 CC06 CF21  
DA12 DB03 DC05 DC16 DE01

【図29】



【図30】



【図31】

